DOCUMENT RESUME

ED 311 870 IR 013 989

AUTHOR Riel, Margaret M.; Levin, James A.
TITLE Learning with Interactive Media: Dynamic Support for Students and Teachers. Interactive Technology Laboratory Report #4.
INSTITUTION California Univ., San Diego, La Jolla. Center for Human Information Processing.
PUB DATE 8 Oct 85
NOTE 27p.
PUB TYPE Information Analyses (070) -- Viewpoints (120)

EDRS PRICE MF01/PC02 Plus Postage.
DESCRIPTORS Authoring Aids (Programing); *Computer Assisted Instruction; Computer Assisted Testing; Computer Games; Computer Simulation; *Computer Software; Educational Innovation; *Individualized Instruction; Instructional Design; *Learning Strategies; Microcomputers; *Technology Transfer; Word Processing; Writing Instruction
IDENTIFIERS *Dynamic Support; *Learner Control

ABSTRACT

The controversy over appropriate educational uses of computers is framed along a continuum based on the amount of support provided to the user. Software programs in which the user's role is to respond in a pre-determined structure (program controlled software) anchors one end of the continuum, while software which empowers the user to create new ways to use the computer (user control software) anchors the other end. Both positions in the controversy are too static. An alternative position is proposed: a process of educational software use in which the amount of assistance provided by the computer is systematically decreased as novices gain expertise. This principle, termed "dynamic support," applies to students learning to write and to teachers learning to incorporate computers into their classrooms. (21 references) (Author/GL)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

X This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

ED311870

Report #4

Learning with Interactive Media: Dynamic Support for Students and Teachers

Margaret M. Riel
James A. Levin
Interactive Technology Laboratory
University of California, San Diego

Barbara Miller-Souviney
Vista Unified School District
Vista, California

IR013989

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

Randall Souviney

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

BEST COPY AVAILABLE

ABSTRACT

The controversy over appropriate educational uses of computers is framed along a continuum based on the amount of support provided to the user. Software programs in which the user's role is to respond in a pre-determined structure (program controlled software) anchors one end of the continuum, while software which empowers the user to create new ways to use the computer (user control software) anchors the other end. With this frame, we argue that both positions in the controversy are too static, and propose an alternative position: a process of educational software use in which the amount of assistance provided by the computer is systematically decreased as novices gain expertise. This principle, which we call "dynamic support," is shown to apply to students learning to write and to teachers learning to incorporate computers into their classrooms.

Computer Educational Software for Students

The role that computers can play in classrooms had been implicitly debated by those who create educational software. The various positions in this debate can be described as points along a continuum of control (Figure 1). At one end of this continuum is the "program controlled" position, in which the initiative for action is contained largely within the computer program being used. At this end of the continuum are a large number of educational software packages that drill or test students on narrowly defined academic tasks. At the other end of the continuum is the learner controlled position, in which the initiative for action rests with the person using the computer. At this "end of the continuum are computer languages and other "tools" that provide students with the power to explore or create new uses of the computer.

Drill & Test CAI	Simulations and educational games	General Purpose Tools (Programming lang., editors)
<---------->		
Computer Controlled	Mixed Control	Learner Controlled

Figure 1: Types of Educational Software for Students

Drill and practice software translates classroom exercises into computer programs adding little more than the ability for students to receive immediate feedback from a machine rather than from the teacher. While advantageous for student motivation, feedback and pacing and teachers' administrative efficiency, drill and practice programs are often criticized as too constraining on both students and teachers (Papert, 1980; Leuhrmann, 1981; Tucker, 1982; Amarel, 1982; Becker, 1985).

General purpose languages (Logo, BASIC, and Pascal), word processing programs designed for students (Bank Street Writer, The Writer's Assistant), graphic and music editors, spreadsheets and data base programs provide access to a broad range of learning activities often with few constraints or directions. When beginning to use such general tools, learners often have more power than they can deal with and little notion of productive strategies or plans for using the tool. Use of these general purpose tools places a burden on teachers to provide the support needed by students, especially for novice users.

In the middle region of the continuum are simulation programs and educational games that share the initiative with the user. In these activities students and teachers are both able to make choices that help frame the educational activity.

Dynamic Support for students

Arguments have been raging over which of these kinds of software represent the best educational use of the computer (Papert, 1980; Johnson, Hanson, & Klassen, 1980; Anderson, Klassen & Johnson, 1981; Luehrmann, 1981. We take a different theoretical perspective on this issue. In our studies of computer use, we have observed that no one position along the continuum is best for all students, or even for the same student at different times. Instead, when students start as novices in some domain, they need a lot of support (from a teacher, from print, and from educational software). As they acquire expertise they no longer need as much support, and when they become experts, they are ready to take over the whole task. Based on our

observation of computer use and this conception of the acquisition of expertise, we have developed an educational design principle called "dynamic support".

Dynamic support refers to the process of systematically decreasing amounts of assistance provided to novices as they progress in expertise and gradually assume parts of the task initially accomplished only by an expert. This notion of dynamic support is derived from the learning principle referred to as the "zone of proximal development" (Vygotsky, 1978; Brown & French, 1979; Griffin & Cole, 1984). In a properly arranged teacher-student-computer environment there is the potential for creating the kind of dynamic support necessary to improve students' learning dramatically . Software which provides dynamic support encourages the progressive development of skill by the learner. Initially, software provides considerable support. As users become more skilled, the support diminishes turning control of the task over to the users. A system of educational software which embodies the principle of dynamic support encourages movement from the left side of the continuum shown in Figure 1 to the right side of the continuum rather than making assumptions about the best location along the continuum (Figure 2).

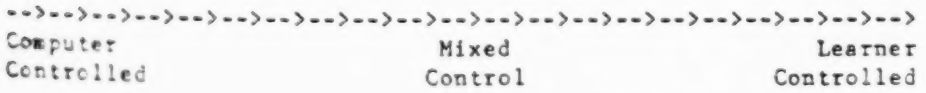

Figure 2: Dynamic Support

The dynamic support principle developed out of research we conducted on how a computer could be used to help students write (Levin, Boruta, &

Vasconcellos, 1983). By examining the pattern of student errors and by observing the social interaction around the computer, we found that students required considerable support as novice writers. Word processing systems can be very powerful tools for writing, but the word processor by itself did not provided a good entry point for students having difficulty with the composing process. Students encountered the problem of "the blank screen" (Levin & Boruta, 1984). As a result the teachers needed to provide instructions on how to use the word processor, but also needed to design supporting writing activities for novice students. These supports were in the form of task cards placed around the computer and textfiles that were used to provide both the directions for a writing activity and the writing itself.

It became apparent that a writing system that would enable teachers to provide this kind of support to the writer on the computer was needed. Learning how to use the many options and commands provided in a word processor needed to follow some initial guidance in learning what to write.

From observations of ways that teachers provided support for writing, Levin developed a system for creating "interactive texts" called the Interactive Text Interpreter (ITI) (Levin, 1982) that embody the concept of dynamic support in the area of writing. Simple commands are used in textfiles to indicate that portion of the text to be presented as instructions, suggestions or examples to the students and what part should become part of the text created by the student. Within this system students are given the option of deciding how many instructions or examples they need to draw upon in the composing process. Teachers can direct students attention to a particular piece of information (such as how many words are

required in a particular line of a poem, or how to describe an event) at the time when the student can best use that information.

Interactive texts are simplified reading and writing environments which can offer the following forms of interaction with the user.

I. Students can make choices among options presented in a menu by:

a) Selecting alternative words or phrases such as choosing one of four possible titles or deciding which character will be featured in a story.

b) Selecting an option which affects the sequencing of text such as choosing a tutorial, asking to see instructions or examples, or determining the outcome of a story by making choices for the characters.

II. Students can be asked to write by:

a) Asking them to provide a word or phrase in a highly constrained "frame" such as "Enter a past tense verb" or "Describe a large animal that is normally found in the desert."

b) Responding to a writing prompt such as "In this first paragraph, tell what you think about the sport, why you selected it, or something that makes it unique." or "Now describe how the game is played, and the goal of each of the players."

c) Composing longer segments of text following general directives such as "Now write your story." or "Enter your Haiku."

At the program-control end of the continuum, "readers" of interactive texts help compose the text by making choices that determine the structure of the essay, or the direction of the plot by selecting from among presented options. At the user-control end of the continuum, interactive texts can place the responsibility for writing in the hands of the students offering only suggestions or examples. A set of such interactive texts can provide students with a range of writing "tools" which vary in the degree of help they provide. We have found that if students use such a range of writing

tools, starting with highly supportive tools and then gradually move to tools requiring them to do more and more of the writing, they gain the skill and confidence to write on their own without help (Mehan, Moll, & Riel, 1985; Miller-Souviney, 1985).

Dynamic Support for Writing Expository Text

Miller-Souviney (1985) has used a set of interactive texts to teach expository writing to fourth and fifth graders. Each of the four expository writing tasks are arranged so that students are able to produce a good example of an essay every time they write. In the first activity, "The Sandwich Prompt," the student "makes" a unique story by choosing among options which are provided throughout the text. Here are the first three choices that the student makes while writing about the art of constructing sandwiches:

SANDWICH PROMPT

Today is

1. Saturday
2. Martin Luther King Jr.'s Birthday
3. Teacher's Workshop Day
4. National Take a Computer to Lunch Day

(Choose 1..4; 0 to end):

***** Type a number then push return ******

---

and I have a day off from school.

My parents are

1. at work
2. climbing Mount Everest
3. eating at a restaurant
4. playing tennis
(Choose 1..4; 0 to end):

***** Type a number then push return ******

---

so I have to make my own

1. breakfast
2. lunch
3. dinner
4. snack
(Choose 1..4; 0 to end):

***** Type a number then push return ******

The following is a story that David produced using this program (Miller-Souviney, 1984). (Underlined text indicates words actually entered by David.)

The Art of Sandwich Construction
by David

Today is National Take a Computer to Lunch Day and I have a day off from school. My parents are playing tennis so I have to make my own lunch. My specialty is that wonder of culinary art, the sandwich! A great French chef, Francois d'Boloney, taught me to make his most secret recipe, The California Kid's Surprise!

First, I need the bread. The recipe calls for two hundred pounds of hamburger buns. I always try to pick good ones, which haven't been painted green.

Next, I put the bread on the counter close to the refrigerator. ...(continues to describe the addition of each of the ingredients)...Gently, I put the top on my magnificent creation and arrange myself in the proper eating position.

This is how I make my Super Secret Recipe Sandwich. When I have a day off school, I never go hungry! Bon appetit!!

The second activity in expository writing involves filling in words and phrases as well as making choices among pre-determined options:

SCHOOL DAY SCHEDULE PROMPT

The name of my school is

?

*** Type, then push CTRL-C when done ****

It is in the town of

?

*** Type, then push CTRL-C when done ****

I am in grade

?

*** Type, then push CTRL-C when done ****

---

and my teacher's name is

?

*** Type, then push CTRL-C when done ****

This "School Day Schedule" prompt provides students with the beginnings of sentences and paragraphs and invites students to complete the ideas begun for them by the expository prompter. In the following composition, the text that Sarah has entered is underlined:

One Day at School

By Sarah

The name of my school is Olive. It is in the town of Vista. I am in grade fourth and my teacher's name is Mrs. Souviney. I have a very busy schedule at school. My class does all sorts of things to make it fun to learn.

The first thing I do is look at the blackboard and copy the daily hand tounge twister. My teacher checks my folder carefully, making sure my writing is nice and neat.

Next, I go to the mat area. It's time for our class meeting where the whole class gets a chance to make announcements and share things. At the end of Class Meeting, my teacher gives us directions for Station Rotation and spelling.

During Station Rotation time, I do different things. I read with my teacher, talk with my friends,sharpen your skills. I practice using my spelling words every week. My favorite spelling activity is news papper knownows because it is the best acttivity.
(...description of school day continues...)

I have a busy schedule every day at school. I'm learning a lot about many different things. I think school this year is grody.

As students increase their writing skills, they take over more and more of the writing activity. In the third expository writing task, the students are asked to describe "how to run a computer." Topic sentences for each paragraph, as well as reminders of what should be contained in each paragraph, are provided. Students enter the introduction, body, and conclusion of their essay.

HOW TO RUN AN APPLE COMPUTER

We are lucky here at Olive School. We have Apple computers. We use them to help us practice our spelling and math, write stories and lots of other things. An Apple computer is easy to run. Just

follow these simple directions.

First, it is important to know the different parts of the computer and what each does.

(Write complete sentences telling the parts of an Apple and what they do.)

?

*** Type, then push CTRL-C when done ****

Next, to start the Apple, several switches need to be turned on.

(In complete sentences, describe where each switch is and what it does.)

?

*** Type, then push CTRL-C when done ****

Students respond to each prompt by entering their own sentences and paragraphs. The following is Margaret's version of how a computer is run (Margaret's text is underlined):

How To Run An Apple Computer

By Margaret

We are luck here at Olive School. We have Apple computers. We use them to help us practice our spelling and math, to write stories and lots of other things. An Apple computer is easy to run. Just follow these simple directions.

First, it is important to know the different parts of the computer and what each does. The moniter shows you what you write and the keyboard has letters to type with.The disk drive is where you put the disk in and the printer is for printing out stories.

Next, to start the Apple, several switches need to be turned on. To turn on the moniter you turn the switch on the bottom right corner.To turn on the keyboard you press the button on the back and to turn the printer on you press the button on the left

side.

(... Continues to describe the steps for using the computer...)

Using a computer isn't all that difficult. If you follow my directions, you should be on your way to being an Apple computer "expert." With a little practice you can use an Apple to help you learn how to do all sorts of things.

In the final activity in the expository writing series, students are asked to think of a topic to write about and given a few structured reminders:

How To

Think of something you know how to do that you could explain to a friend. For example, you may know how to wash dishes, run a computer or clean your room.

Remember:

1) The first paragraph is your introduction and will tell something about the activity.

2) The second paragraph will tell the steps it takes to do it.

3) The last paragraph is your conclusion and can tell about how you feel when you finish the activity or when you will do it again.

In this, the fourth phase of the expository writing activity, students are now responsible for producing complete texts on their own. Here are two stories produced by Lorenzo and Armando from Miller-Souviney's (1985) classroom (text they entered is underlined):

How To Make Money
by
Lorenzo

If you want to make money, you must make it right now. If you want to know why, you can say because you would want to help your family, or something

else.

To make money, the easiest way could be gathering cans. Cans can be everywhere, so when you are walking and you see a can, stop and pick it up.

When you have gathered at least 500 cans, give them to a store. they will give you money for the cans. Then if you want to do it again, do it and you will have more money.

Throughout these four activities, the goal is for the quality of the writing to remain constant. As the degree of the participation by the learner increases, the amount of support provided by the computer decreases (figure 3).

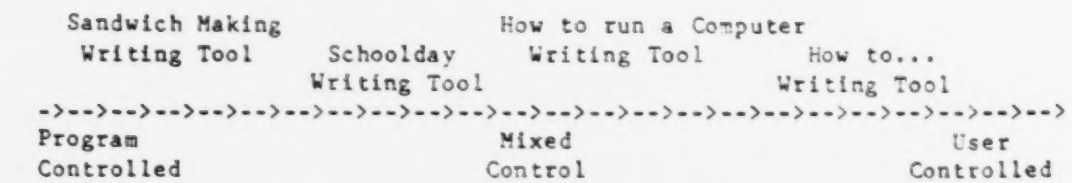

Figure 3: Dynamic Support for Expository Writing

Miller-Souviney used this sequence of computer activities to teach expository writing to her class of 28 4th and 5th graders over a four month period. The initial writing skill of each student was assessed by two writing assignments, one written on paper and the other on the computer. After using all four of the Expository Writing Tools, the students' writing was again assessed on and off the computer.

Writing quality and fluency improved significantly over the four month period of instruction in which dynamic support was provided by the computer programs. The quality of the students' writing was evaluated by an holistic scoring rubric (Grubb, 1981). Both pre- and post-writing samples were assessed

by four independent scorers each of which used a 4 point scale producing a total score range of 0 to 16. The average student score on the pretest was 7.5 on the pencil and paper test and 7.7 on the computer test. The average posttest score was 10.8 for the pencil and paper test and 10.4 for the computer test. A test of significance was computed comparing selected pre and post measures using the non-parametric Wilcoxon Matched-Pairs Signed-Ranks Test (WST) for differences between related samples (Siegel, 1956). The results show that the quality of the students' expository writing improved significantly ($p < 0.005$).

	Pretest	Posttest
Paper/pencil (n=25)		
Holistic score (0-16)	7.5	10.8 *
Total word count	57.7	97.5 *
Unique word count	34.4	55.4 *
Computer (n=25)		
Holistic score (0-16)	7.7	10.4 *
Total word count	70.8	81.0
Unique word count	38.9	45.7

* significant $p<0.005$ (WST)

Word count data indicated an improvement in fluency rates as well. The average total word count increased by nearly 40 words and and the unique word count by over 20 words between the pre- and posttests written on paper. This improvement was also significant. The fluency rates also increased on the computer tests but they were not significant.

Research on the use of word processors in classrooms often report an increase in the length of students' writing with no increase in the quality of the writing (Daiute, 1982; Levin, Boruta, & Vasconcellos, 1983). Similarly, Miller-Souviney found that the length of students' pretests written on paper and on the computer demonstrate the highly motivating nature

of the computer: students wrote longer essays. However these essay were not necessarily of higher quality. The use of interactive writing tools arranged to provide dynamic support in this study resulted in increases in quality as well as in length.

By arranging learning environments in which computer-based support was gradually removed, students gained control of expository writing by gradually assuming the parts of the task initially accomplished by the computer. Dynamic support provided by the microcomputer subordinated the students' concern for the mechanics of writing to the process of writing, resulting in improved quality and fluency.

In a classroom in which this sequence of activities was used in poetry as well as in other forms of writing, some students reached a point in which they asked if they could use the word processor directly to write their poems (Mehan, Moll, & Riel, 1985). They were ready to give up the support of Interactive Tools for the increased editing capability of the word processor. This development of independent skill and control over the computer is the goal of "dynamic support."

Educational Software for Teachers

So far we have been discussing the relationship of the learner to the software and the need for dynamic support in the learning process. These same relationships hold if we change the focus to the teacher and the teaching process.

Teachers are being placed in a very difficult position with the recent availability of computers for education. They are being asked to introduce their students to this new technology and to prepare them for using computers in a rapidly changing world. The computer is often seen as a self-contained system which will produce revolutionary new forms of learning and teaching when placed in the hands of students and teachers (Papert, 1980). But like many other educational innovations, the computer is only a tool and its effectiveness will depend on how it is used. Teachers often find themselves in the position of deciding what should be done with the computer in the classroom with little preparation for, or knowledge about, teaching with computers.

Given this situation, it is not surprising that some teachers make the computer itself the object of study as students as well as teachers try to discover what it means to become "computer literate." Teaching students computer literacy is difficult because of the rapid rate of change in computer technology. The machines of today will not be the machines of tomorrow. Learning how to use a computer is not the same thing as learning how computers work. Teaching students the rudiments of programming in the general purpose languages which are now available rarely provides students with enough control over the computer to make it serve their present and future purposes. These languages are likely to be replaced with more powerful special purpose languages in the future, making the mastery of these computer languages obsolete.

An alternative approach for using computers in classrooms is to integrate them into the school curriculum, to use computers as tools to teach

subject matter such as reading, writing, and math (Mehan, Moll, & Riel, 1985; Levin & Souviney, 1984). But again there are a range of conceptions of how computers can be used by teachers to help students acquire basic skills. The same continuum from program control to user control that we presented earlier in this paper (see figure 1) can be used to describe the relationship between teachers and educational software (see Figure 4).

Static frames with fixed content	Lesson frames with content added	Programming Languages & Authoring Systems
<----------	----------	---------->
Software Controlled	Mixed Control	Teacher Controlled

Figure 4: Educational Software for Teachers

At the software-controlled end of the continuum, there are programs that have been developed to be used "as is" with little or no need to change them. This software is easy to use by a teacher who is new to computers, but it is often not possible to adapt it to the changing needs of the students or teacher. Frequently these programs are worksheets implemented on the computer to be used with programs that are highly structured for both teaching and learning. Such programs require teachers to adapt their lessons to the content presented.

In the center of the continuum are a variety of educational software packages that provide a sequence or frame in which teachers add their own content (Missing Links, Square Pairs, Game Show). These programs share the initiative with teachers making it possible for them to modify and adapt computer materials in ways that assure a better match between the computer use and the instructional goals of the teacher. At the teacher-controlled

end of the continuum, there are programming languages (BASIC, Pascal, Logo) and authoring languages (Pilot) that enable teachers to create materials that are congruent with their teaching objectives.

Teachers often complain about the lack of quality in currently available educational software. Some teachers believe that the best way to deal with this problem is to gain control over the computer by learning to create their own software. To achieve this goal they sign up for evening courses in programming, usually in a general purpose computer language like BASIC. Such efforts often discourage teachers. The skills necessary to write the quality programs that these teachers want in their classrooms are difficult to acquire in the limited "free" time available to teachers.

Dynamic Support for Teachers

One approach to this dilemma that we've found effective is to consider teachers as learners who need the same type of "dynamic support" for integrating computer instruction with classroom lessons as we have been describing for students learning a particular form of writing. Just as a word processing system was not the best entry point for all students learning to write, programming is also not the best entry point for all teachers who want to integrate computers into their curriculum.

By working closely with teachers, we have found that novice computer-users were frequently overwhelmed by the power of general purpose programming languages or authoring systems. At first these teachers gravitated toward software that could be used by their students with minimal teacher

modification. As they became familiar with the computer and understood the problems and successes of their students interacting with computers, however, they become more critical of software that they could not control. The experience of modifying programs motivated teachers to find ways to create their own programs.

A solution to the problem is to develop software systems that incorporates the principle of dynamic support for teacher as programmer (figure 4).

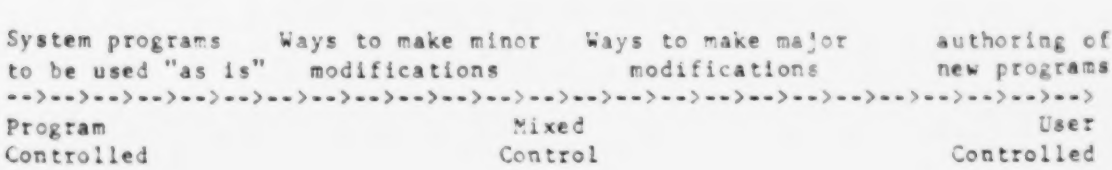

Figure 4: Dynamic Support for Teachers

With software systems that provide "dynamic support" teachers can find materials that are "classroom ready" to encourage immediate use by novice teachers with no or minimal input. These initial activities provide teachers with confidence in using the computer as a tool in the classroom in the same way reluctant writers begin by using writing tools that create text through the selection of options.

Once teachers have used the programs successfully, they become interested in ways to modify or personalize the software for the special interests and needs of their students. The effect of modifying programs and the experiences of success at this level, provides the motivation to consider the creation of new programs.

Ultimately, the goal of such software systems is to develop teachers' expertise so that they can create their own educational uses of the computer. But we are suggesting an alternate model to the one that has teachers learning an authoring system one step at a time. Instead we have discovered that an effective strategy is to have teachers begin by using models of the type of software that can be created with a system. Then they look inside the model, see how it was constructed and how it can be changed, first in trivial ways but progressing to more serious modification. Working with programming tools that help create new programs they can watch the placement of symbols as they create software. The last step would be to master the authoring language.

Close collaboration between researchers, programmers and teachers has evolved in such an authoring system for language arts. The ITI system (Levin, 1982), used to create the Writing Tools described in the first part of this paper, was initially designed as an authoring system for teachers to create language arts software. At first we were disappointed when teachers did not immediate use this tool to create new software. We discovered that teachers were initially more excited about using the example Writing Tools than they were about creating their own software.

Once they began using these tools, some of the teachers did begin to modify them, gradually making more and more substantial changes, to adapt them to their teaching situation. For example, in some of the story-making programs, some teachers changed character names and settings to ones that were familiar to the students. Other teachers changed the instructional sequence either making it more specific for younger students, or removing

help that was too detailed for older students.

The opportunity to modify existing programs encouraged teachers to think of new ways the computer could be used to help students acquire important academic skills. For example, Miller-Souviney had no prior programming experience when she created the the Expository Writing Tools to provide dynamic support for expository writing (Miller-Souviney, 1985). Other teachers contributed to the development of a newspaper writing tool to provide students with at range of support for writing different types of newspaper articles (Riel, 1983; Levin, Riel, Rowe, & Boruta, 1985). A bilingual teacher learned to create bilingual Writing Tools to help her students with reading and writing skills (Mehan, Moll, & Riel, 1985). Our continuing experience with teachers using computers suggests that teachers can benefit from the same sequence of dynamic support that works with their students.

This movement from using previously developed programs, to modifying programs, to creating new programs and new uses for the system is what we mean by dynamic support. Like students, teachers begin by using the computer in productive ways while gaining the confidence and expertise to better integrate the computer with regular classroom instruction. In this way, teachers can be assured that the activities that occur on the computer are related to those that take place off the computer in classroom instruction.

Conclusion

From working collaboratively with students and teachers we have found that effective use of the computer in classrooms requires software that provides a dynamic range of support. We have described here how such a system operates from two different perspectives. The first perspective was that of a student acquiring writing skills through working with a set of computer writing tools that provided dynamic support.

Just as students benefit from dynamic support in the learning process, we also found that teachers need dynamic support in the process of integrating computer instruction with the teaching goals in academic areas. General purpose computer languages and authoring systems are not the optimal entry point for teachers who seek to use computers for instruction. Instead, we found that a set of materials that can be used immediately, then modified and finally recreated, can provide the support that helps teachers take an active role in the use and development of educational software for their classrooms.

References

Amarel, M. (1983) Classrooms and computers as instructional settings. Theory Into Practice, 22(4): 260-272.

Anderson, R, Klassen, D, & Johnson, D. (1981). In defense of a comprehensive view of computer literacy: A reply to Leuhrmann. Mathematics Teacher, 74, 687-690.

Becker (1985)

Brown, A. & French, L. (1979) The Zone of Proximal Development: Implications for Intelligence Testing for the Year 2000. University of Illinois: Center for the Study of Reading, 1979.

Daiute, C. (1982) Wordprocessing, can it make even good writers better? Electronic Learning March/April 19-31.

Grubb, M. (1983). Using Holistic Evaluation. Encino, CA: Glenco Publishing Co.

Griffin, P & Cole, M. (1984) Current Activity for the Future: The Zo-ped. B. Rogoff & J.V. Wertsch (Eds.), Children's Learning in the "Zone of Proximal Development." New Directions for Child Development. No. 23, pp. 45-63. San Francisco: Jossey-Bass, 1984.

Johnson, D., Anderson, R, Hasen, T & Klassen, D. (1980). Computer literacy--What is it? Mathematics Teacher 73, 91-96

Laboratory of Comparative Human Cognition. (1982). A model system for the study of learning difficulties. The Quarterly Newsletter of the Laboratory of Comparative Human Cognition, 4, 39-66.

Levin, J. A. (1982). Microcomputers as interactive communication media: An interactive text interpreter. The Quarterly Newsletter of the Laboratory of Comparative Human Cognition, 4, 34-36.

Levin, J. A., & Boruta, M. J. (1983) Writing with computers: you get exactly the right amount of space. Theory Into Practice, 22(4): 291-245.

Levin, J. A., Boruta, M. J., & Vasconcellos, M. T. (1983). Microcomputer-based environments for writing: A Writer's Assistant. In A. C. Wilkinson (Ed.), Classroom computers and cognitive science. New York: Academic Press.

Levin, J. A., Riel, M. M., Rowe, R. D., & Boruta, M. J. (1985). Muktuk meets jacuzzi: Computer networks and elementary school writers. In S. W. Freedman (Ed.), The acquisition of written languages: Revision and response. Hillsdale, NJ: Ablex.

Levin, J. A., & Souviney, R. Computer literacy: A time for tools. The Quarterly Newsletter of the Laboratory of Comparative Human Cognition,

5, 45-46.

Luehrmann, A. (1981). Computer literacy: What should it be? Mathematics Teacher, 74, 680-686.

Mehan, H., Moll, L., & Riel, M. M. (1985). Computers in classrooms: A quasi-experiment in guided change. (Final Report to NIE). La Jolla, CA: Teacher Education Program.

Miller-Souviney, B. (1985). Computer-supported tools for expository writing: One computer and 28 kids. Master's Thesis: University of California, San Diego.

Papert, S. (1980). Mindstorms. New York: Basic Books.

Riel, M. M. (1985). The Computer Chronicles Newswire: A functional learning environment for acquiring literacy skills. Journal of Educational Computing Research, 1, 317-337.

Siegel, S. (1956). Non parametric statistics for the behavioral sciences. New York: McGraw-Hill Book Co.

Tucker, M. (1983) Solving achievement problems in bits and bytes. Education Week, April 27, 19-20.

Vygotsky, L. (1978) Mind in society: The development of higher psychological processes. M. Cole, V. John-Steiner, S. Scribner & E. Souberman (Eds.), Cambridge: Harvard University Press.